

## BRT Sys AN-003 LDSBus Python SDK User Guide

[Home](#) » [BRT Sys](#) » BRT Sys AN-003 LDSBus Python SDK User Guide 



Application Note  
BRTSYS\_AN\_003  
LDSBus Python SDK on IDM2040 User  
Guide  
Version 1.2  
Issue Date: 22-09-2023

### Contents

- [1 AN-003 LDSBus Python SDK](#)
- [2 Introduction](#)
- [3 Credits](#)
- [4 Getting Started with IDM2040](#)
- [5 Thorny Python IDE – Installation/Setup Instructions](#)
- [6 Contact Information](#)
- [7 Documents / Resources](#)

## AN-003 LDSBus Python SDK

This document provides information about how to setup and use the LDSBus Python SDK on IDM2040. Use of BRTSys devices in life support and/or safety applications is entirely at the user's risk, and the user agrees to defend, indemnify, and hold BRTSys harmless from any and all damages, claims, suits, or expense resulting from such use.

## Introduction

This document describes how to use IDM2040 with LDSU circuitry example including the Installation procedure for Thorny Python IDE and steps to execute LDSU circuitry examples.

The Python SDK will run on IDM2040 with appropriate LDSBus interface. IDM2040 has built-in LDSBus interface and can supply up to 24v to the LDSBus. More information on the IDM2040 is available at <https://brtsys.com>.

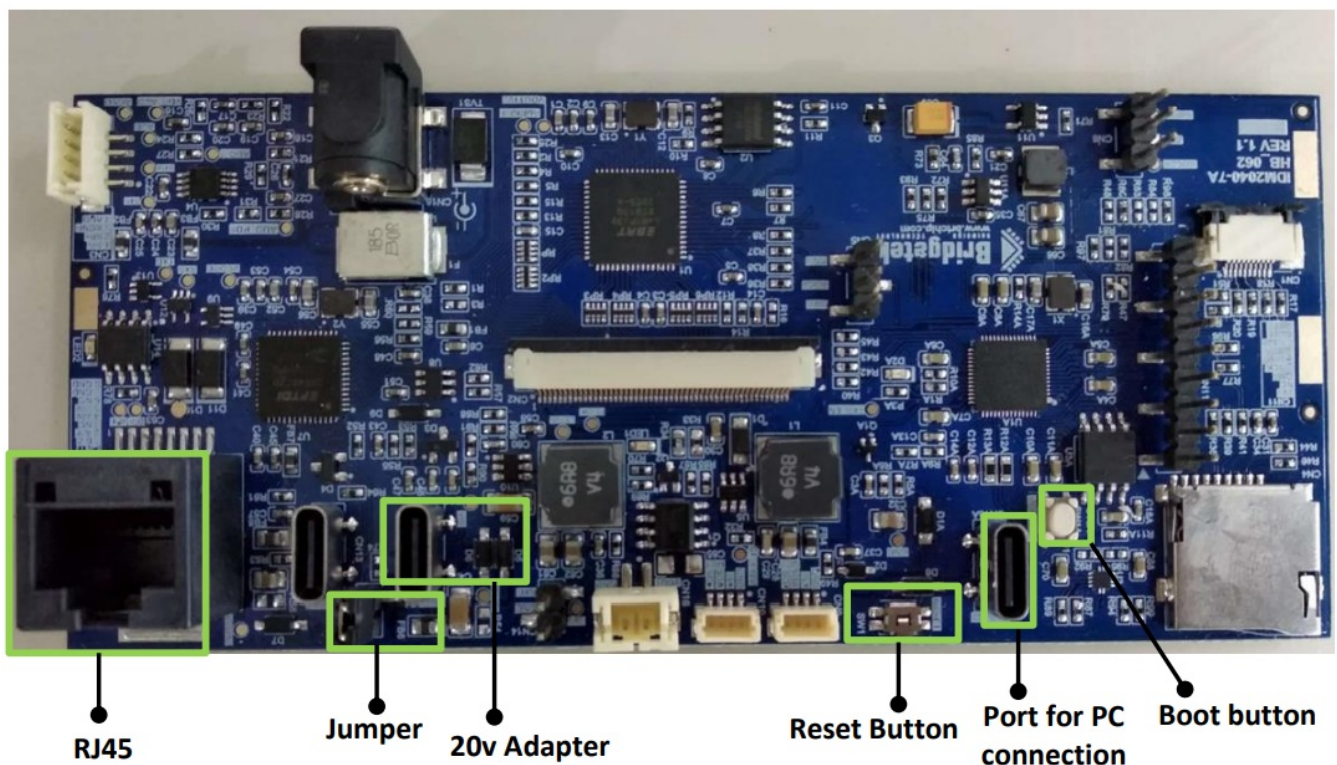
## Credits

Open-Source Software

- Thorny Python IDE: <https://thonny.org>

## Getting Started with IDM2040

### 3.1 Hardware Overview

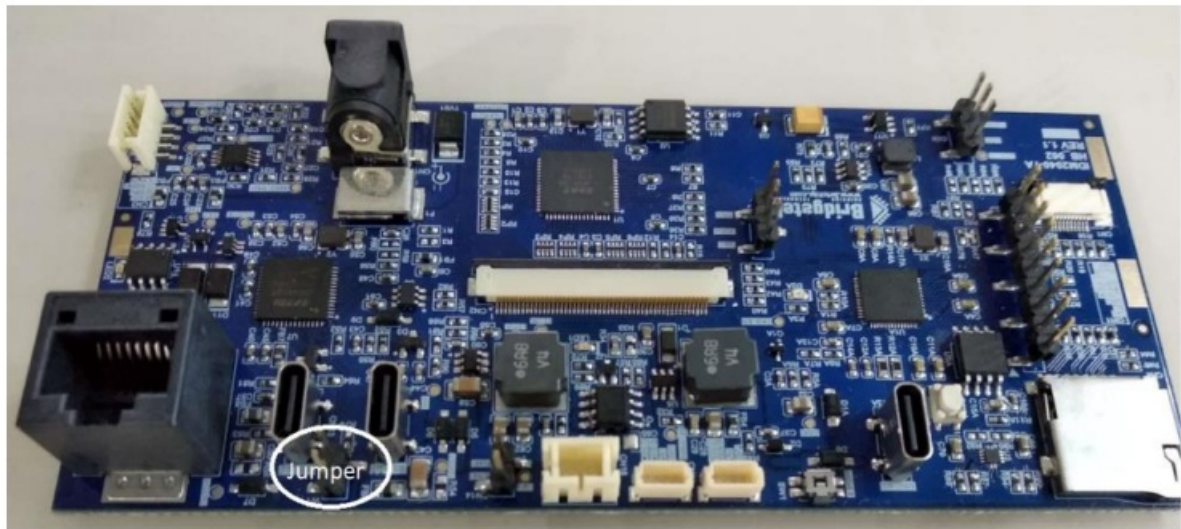


**Figure 1 - IDM2040 Hardware Features**

### 3.2 Hardware Setup Instructions

Follow these steps to setup the IDM2040 Hardware Setup –

- a. Remove the Jumper.

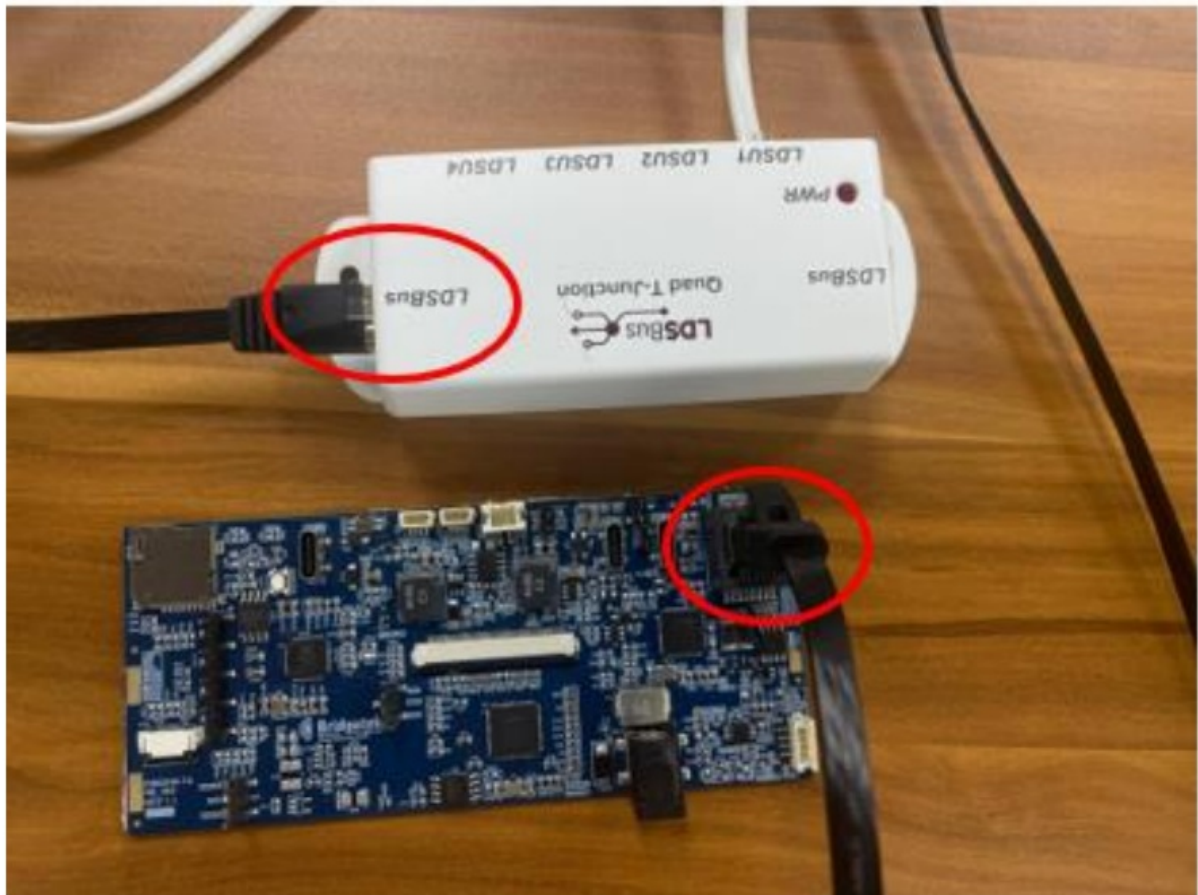


b. Connect the LDSU module to Quad T-Junction.



c. Using RJ45 cable, connect Quad T-Junction to IDM2040 RJ45 connector.





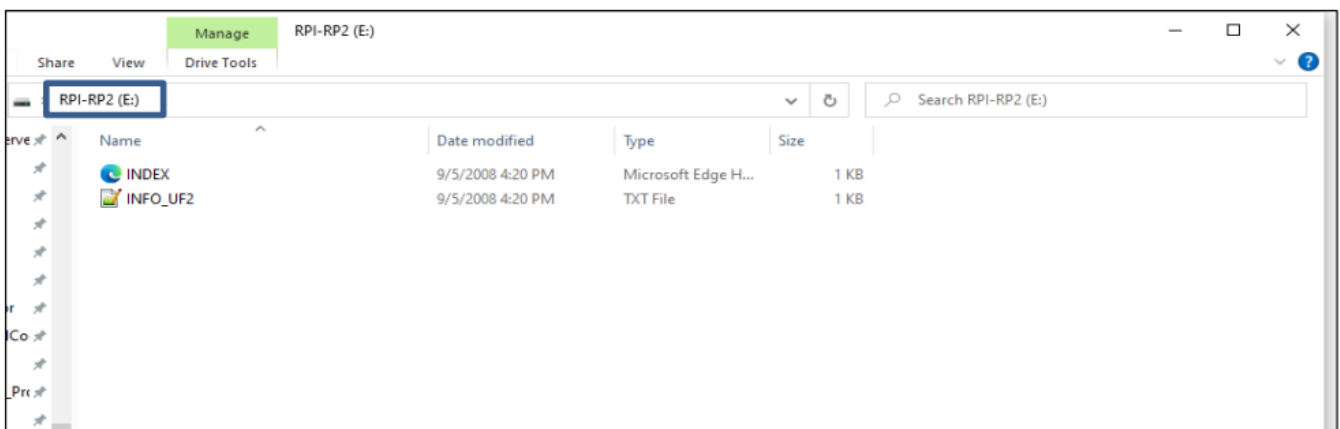
d. Connect the 20v supply adapter using a USB-C cable to the USB-C port on the IDM2040.



- e. Turn on the 20v adapter using the AC power supply.
- f. Connect IDM2040 to PC using Type-C cable.



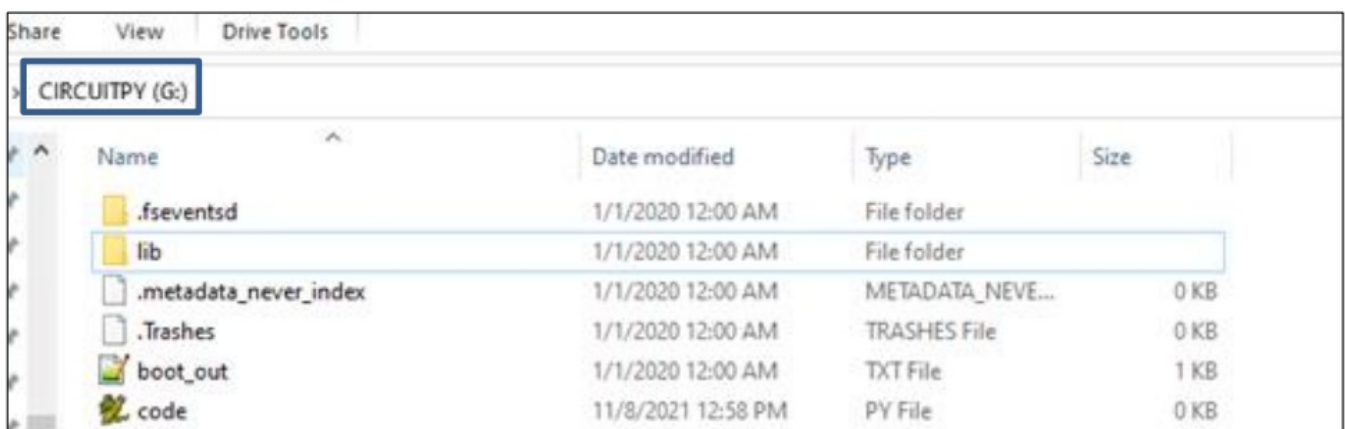
g. Press the Boot button the IDM2040 board; Hold it for a few seconds and release it after resetting the board. Windows will open a drive named “RP1-RP2”.



h. In the given example package, there must be an “.uf2” file, copy the file and paste it into “RP1-RP2” drive.

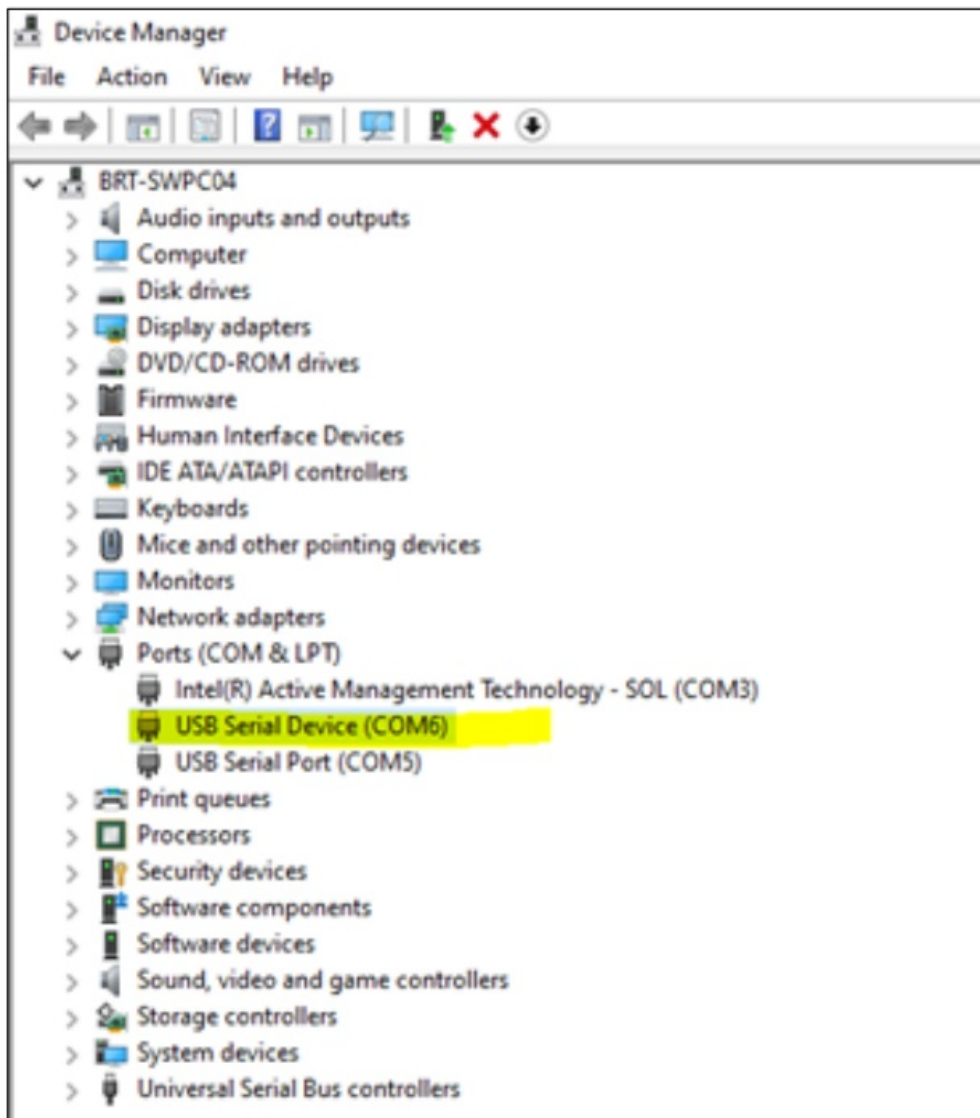
Name	Date modified	Type	Size
SAMPLES	11/11/2021 1:40 PM	File folder	
IDM2040_FIRMWARE.uf2	11/11/2021 1:36 PM	UF2 File	1,512 KB

i. Upon copying the “.uf2” file to “RP1-RP2”, the device will reboot automatically and again will appear as a new drive, such as “CIRCUITPY”.



The “code.py” is the main file which runs every time the IDM2040 is reset. Open this file and delete any content inside of it before saving.

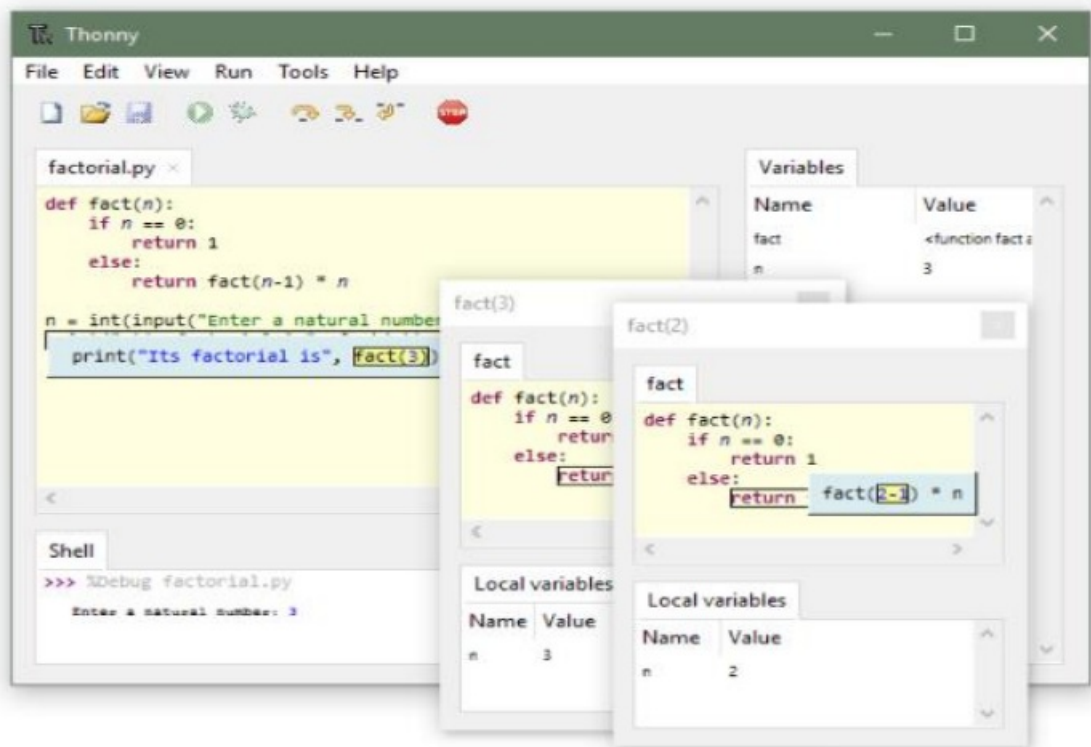
j. The COM port for this device will appear in Device Manager. Here is an example screen showing the IDM2040's COM Port as COM6.



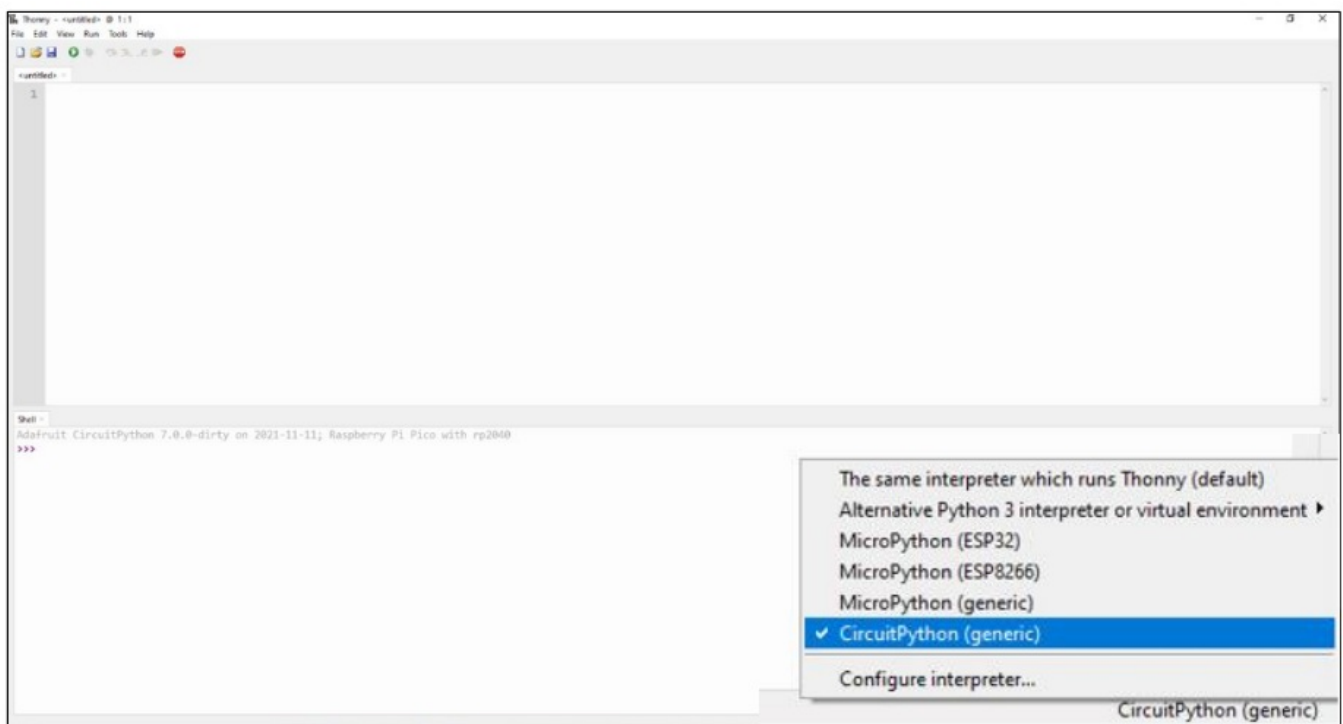
## Thorny Python IDE – Installation/Setup Instructions

Follow these steps to install and setup Thorny Python IDE –

- Download the Thorny Python IDE package from <https://thonny.org/>.
- Click **Windows** to download the windows version.

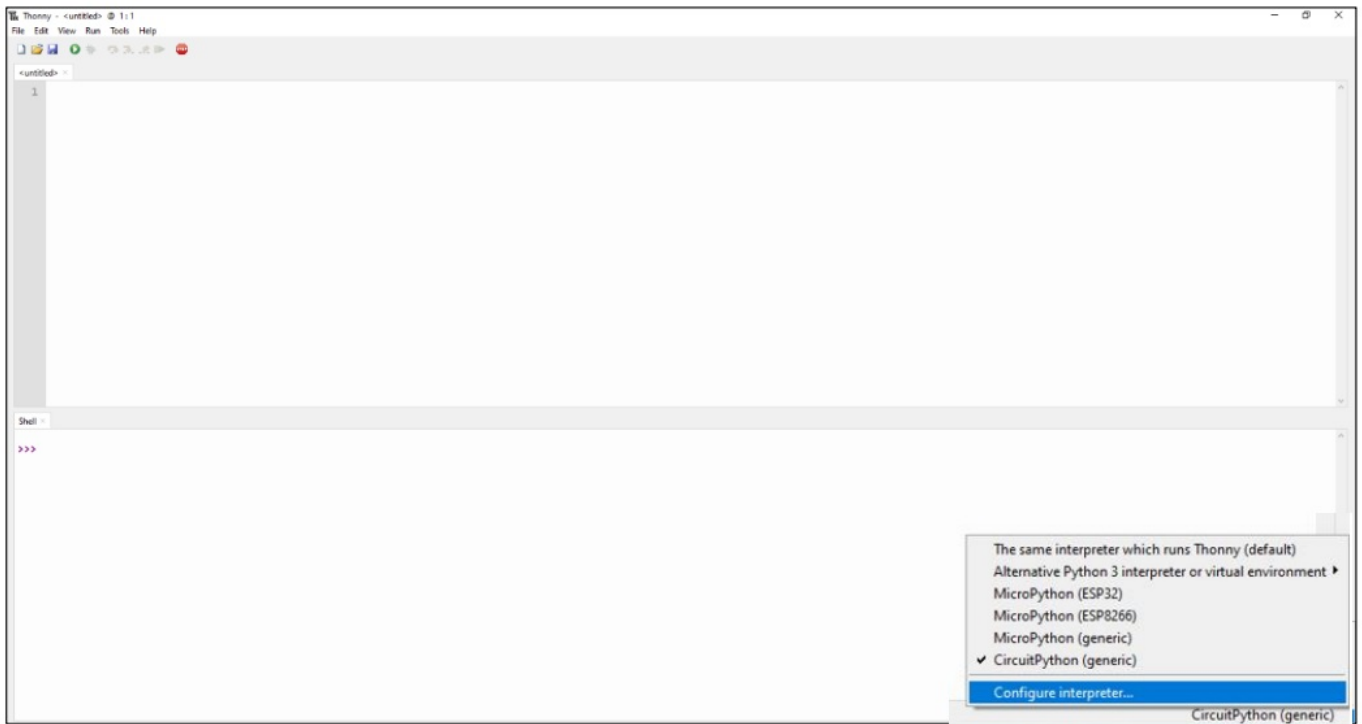


- c. Upon downloading the application, complete the installation by clicking the executable file (.exe) and following the installation wizard. Upon completing the installation, open the Thonny Python IDE from the Windows Startup.
- d. To open the Properties, click the left mouse button at the right bottom corner. Select "Circuit Python (generic)".

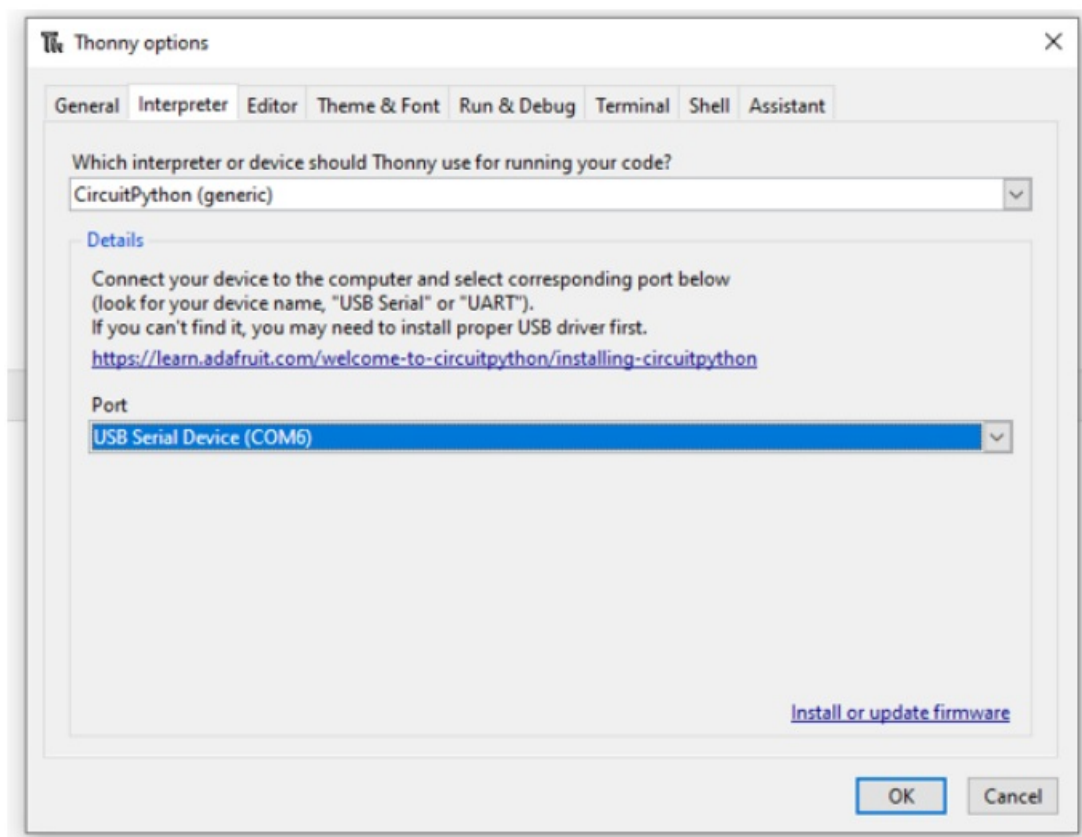


- e. Click "Configure Interpreter..."





f. Click on the Port drop down and select the port appeared for IDM2040 in device manager after connecting. In this example screenshot COM port appeared as COM6. Click **[OK]**.



g. Thonny will report the device information at the interpreter prompt (“Adafruit Circuit Python 7.0.0-dirty on 2021-11-11; Raspberry Pi Pico with rp2040”) if the device port is correct.

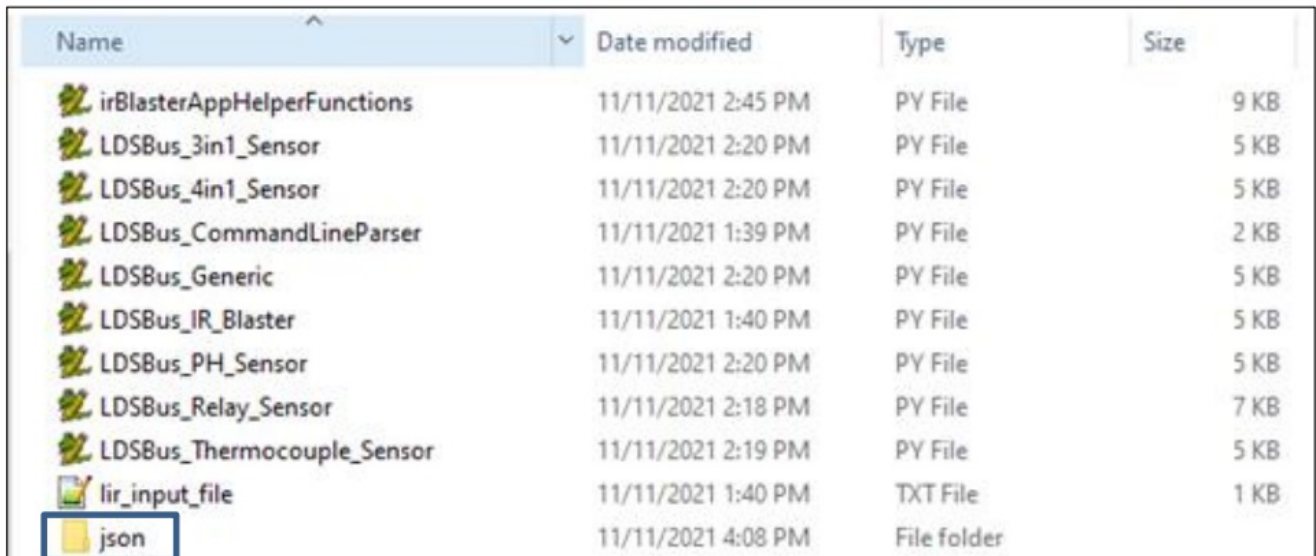
```
Adafruit CircuitPython 7.0.0-dirty on 2021-11-11; Raspberry Pi Pico with rp2040
>>>
```

Procedure to run LDSU Circuitry Sample Example using Thonny



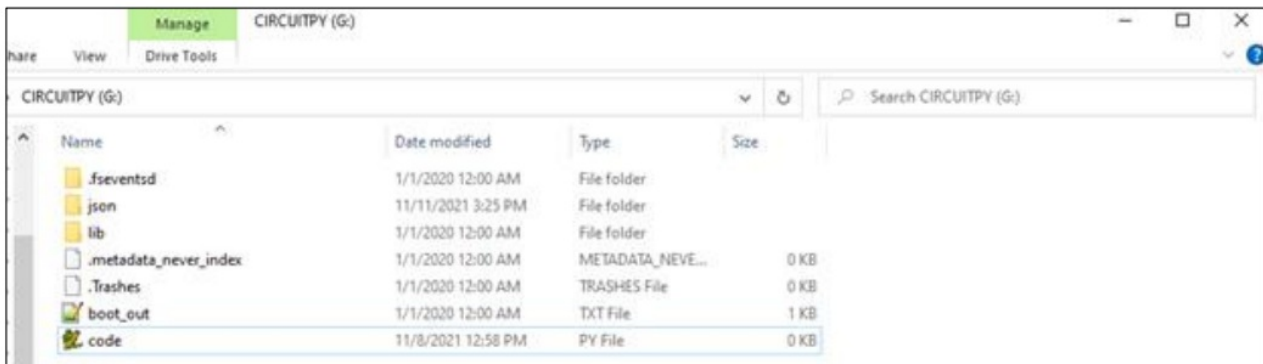
Follow these steps to run the LDSU circuitry sample example –

a. Open the sample package file. As part of the sample package there is a folder by name “son” which contains various sensor son file.



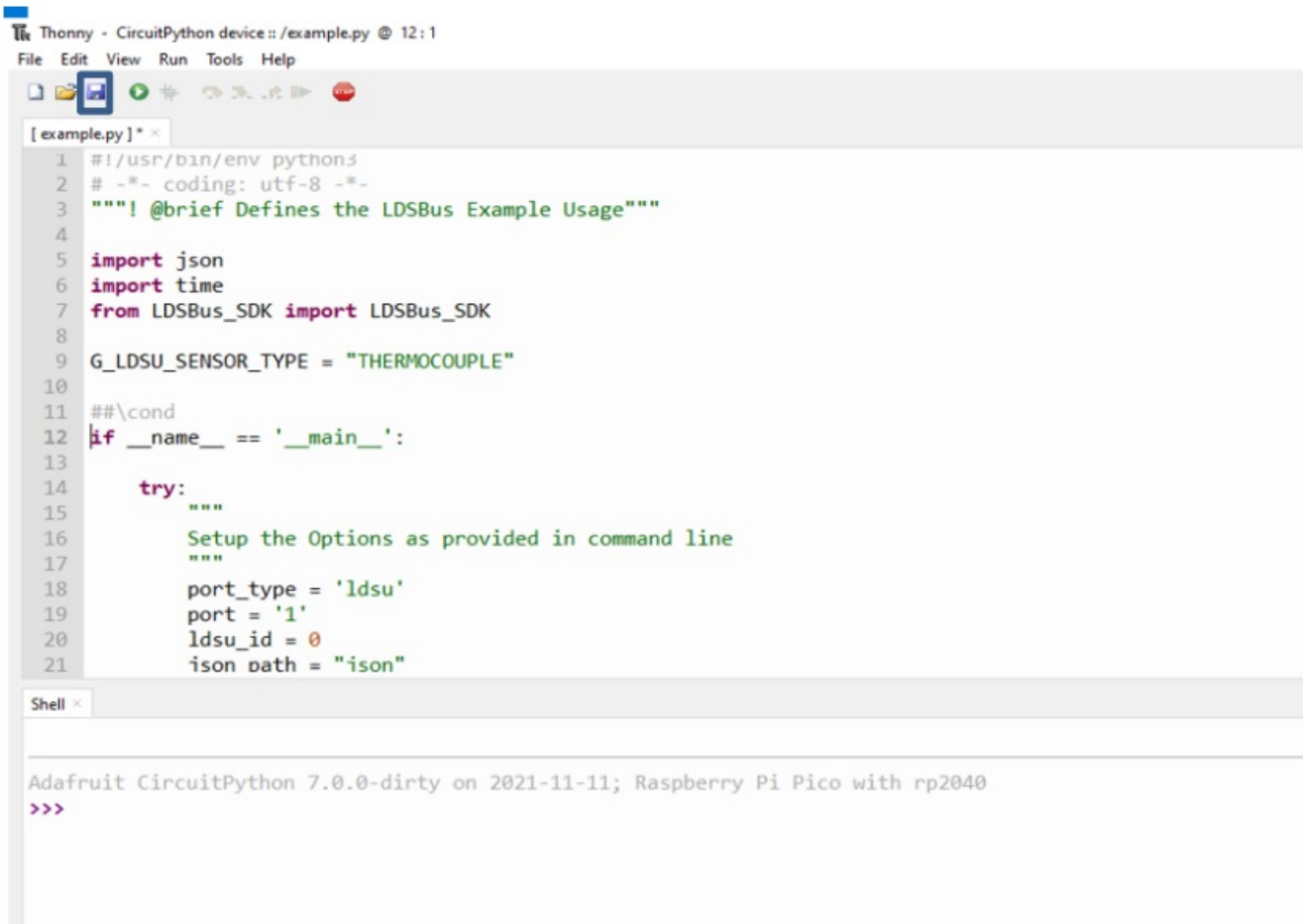
Name	Date modified	Type	Size
irBlasterAppHelperFunctions	11/11/2021 2:45 PM	PY File	9 KB
LDSBus_3in1_Sensor	11/11/2021 2:20 PM	PY File	5 KB
LDSBus_4in1_Sensor	11/11/2021 2:20 PM	PY File	5 KB
LDSBus_CommandLineParser	11/11/2021 1:39 PM	PY File	2 KB
LDSBus_Generic	11/11/2021 2:20 PM	PY File	5 KB
LDSBus_IR_Blaster	11/11/2021 1:40 PM	PY File	5 KB
LDSBus_PH_Sensor	11/11/2021 2:20 PM	PY File	5 KB
LDSBus_Relay_Sensor	11/11/2021 2:18 PM	PY File	7 KB
LDSBus_Thermocouple_Sensor	11/11/2021 2:19 PM	PY File	5 KB
lir_input_file	11/11/2021 1:40 PM	TXT File	1 KB
json	11/11/2021 4:08 PM	File folder	

b. Copy and paste the “json” folder to the “CIRCUITPY” storage device.

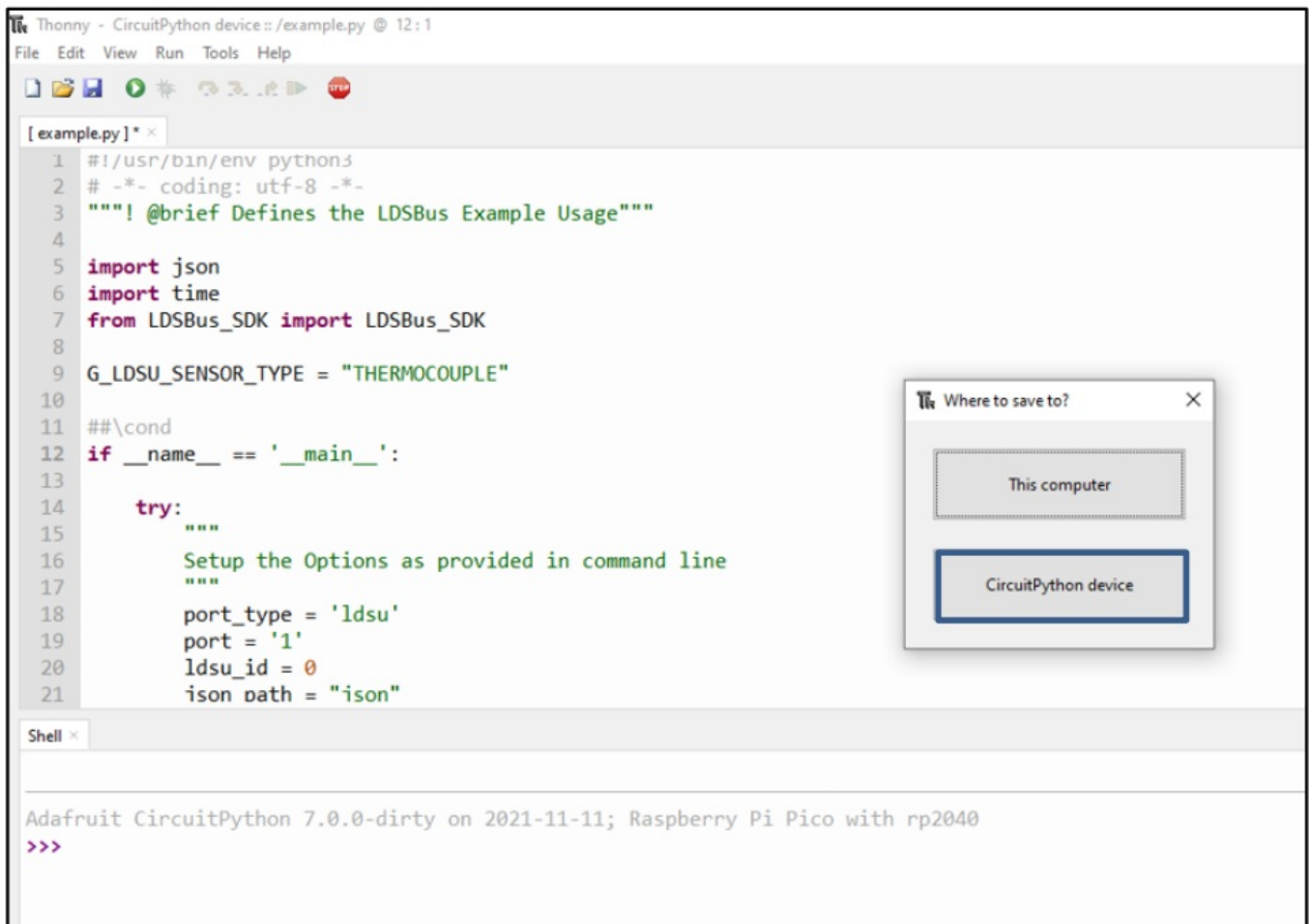


Name	Date modified	Type	Size
.fseventsd	1/1/2020 12:00 AM	File folder	
json	11/11/2021 3:25 PM	File folder	
lib	1/1/2020 12:00 AM	File folder	
.metadata_never_index	1/1/2020 12:00 AM	METADATA_NEVE...	0 KB
.Trashes	1/1/2020 12:00 AM	TRASHES File	0 KB
boot_out	1/1/2020 12:00 AM	TXT File	1 KB
code	11/8/2021 12:58 PM	PY File	0 KB

c. Open any given example using a text editor such as notepad ++ and copy it to the Thorny Editor and save it. For example, open “LDSBus\_Thermocouple\_Sensor.py” and copy/paste on Thorny Editor. Click **[Save]**.

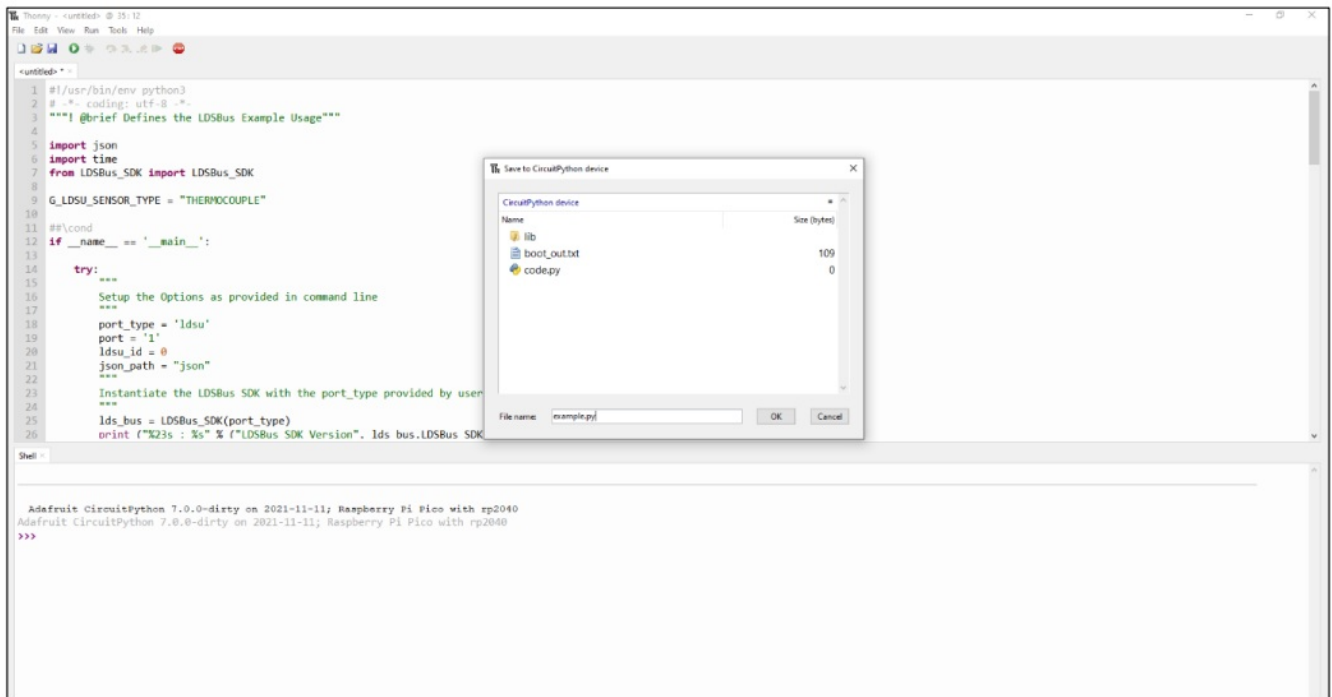


d. Upon clicking [Save], a "Where to save to?" dialog box will be displayed. Click and select Circuit Python device.

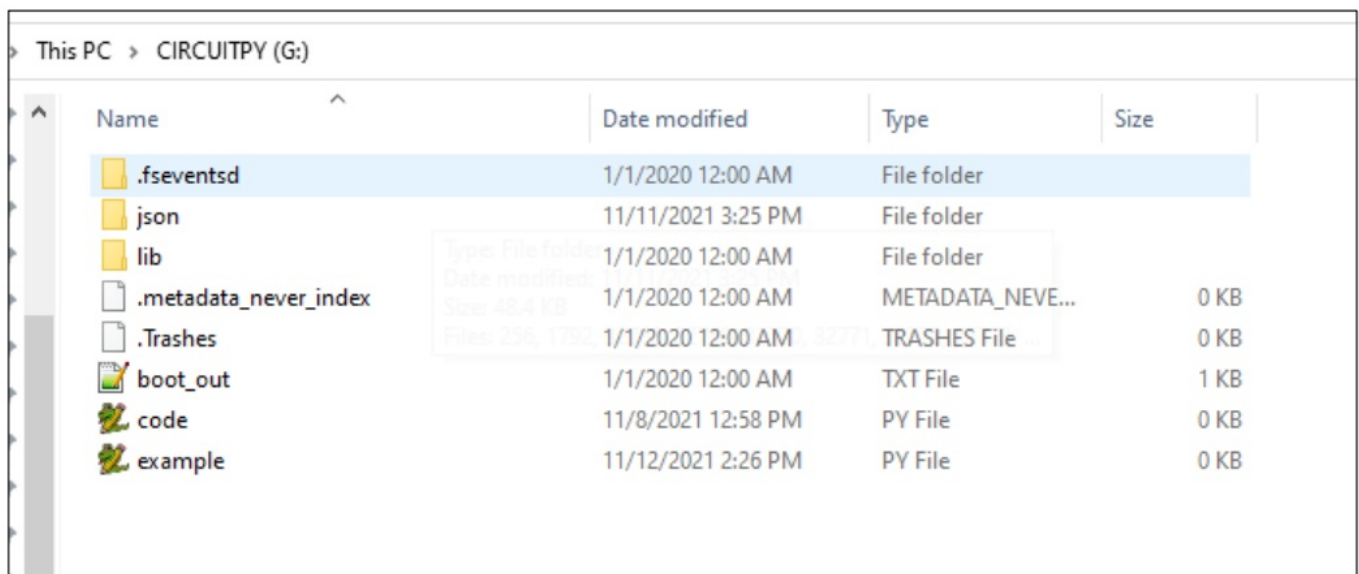


e. Enter a file name and click [OK].

**Note:** When sample code is saved to “code.py” then every time it reboots, it will begin running “code.py”. To avoid this, specify a different name.



f. The file will be saved to “CIRCUITPY” drive.



g. To run the example from Thorny Editor, click  (Run current script).



```
Thonny - CircuitPython device :: /example.py @ 12:1
File Edit View Run Tools Help

[example.py]* x
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 """! @brief Defines the LDSBus Example Usage"""
4
5 import json
6 import time
7 from LDSBus_SDK import LDSBus_SDK
8
9 G_LDSU_SENSOR_TYPE = "THERMOCOUPLE"
10
11 ##\cond
12 if __name__ == '__main__':
13
14     try:
15         """
16         Setup the Options as provided in command line
17         """
18         port_type = 'ldsu'
19         port = '1'
20         ldsu_id = 0
21         ison path = "ison"

```

```
Shell x
Adafruit CircuitPython 7.0.0-dirty on 2021-11-11; Raspberry Pi Pico with rp2040
>>>
```

h. The Circuitry LDSU example will run to scan the bus and start reporting the sensor data.



Thonny - CircuitPython device :: /example.py @ 21:27

File Edit View Run Tools Help

```

[ example.py ] x
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  """! @brief Defines the LDSBus Example Usage"""
4
5  import json
6  import time
7  from LDSBus_SDK import LDSBus_SDK
8
9  G_LDSU_SENSOR_TYPE = "THERMOCOUPLE"
10
11  ##\cond
12  if __name__ == '__main__':
13
14      try:
15          """
16          Setup the Options as provided in command line
17          """
18          port_type = 'ldsu'
19          port = '1'
20          ldsu_id = 0

```

Shell x

```

          TYPE : LDSBus Thermocouple Sensor
          Thermocouple : 28.13    C
          =====
          =====SENSOR DATA=====
          UUID : LS01010114162113360
          TYPE : LDSBus Thermocouple Sensor
          Thermocouple : 28.13    C
          =====
          =====SENSOR DATA=====
          UUID : LS01010114162113360
          TYPE : LDSBus Thermocouple Sensor
          Thermocouple : 28.00    C
          =====
          =====SENSOR DATA=====
          UUID : LS01010114162113360
          TYPE : LDSBus Thermocouple Sensor
          Thermocouple : 28.13    C
          =====
          =====SENSOR DATA=====
          UUID : LS01010114162113360
          TYPE : LDSBus Thermocouple Sensor
          Thermocouple : 28.00    C
          =====

```

j. Remember to copy the following files – “irBlasterAppHelperFunctions” and “lir\_input\_file.txt” before trying the LDSBus\_IR\_Blaster.py example.

Name	Date modified	Type	Size
json	11/11/2021 4:08 PM	File folder	
irBlasterAppHelperFunctions	11/11/2021 2:45 PM	PY File	9 KB
LDSBus_3in1_Sensor	11/11/2021 2:20 PM	PY File	5 KB
LDSBus_4in1_Sensor	11/11/2021 2:20 PM	PY File	5 KB
LDSBus_CommandLineParser	11/11/2021 1:39 PM	PY File	2 KB
LDSBus_Generic	11/11/2021 2:20 PM	PY File	5 KB
LDSBus_IR_Blaster	11/11/2021 1:40 PM	PY File	5 KB
LDSBus_PH_Sensor	11/11/2021 2:20 PM	PY File	5 KB
LDSBus_Relay_Sensor	11/11/2021 2:18 PM	PY File	7 KB
LDSBus_Thermocouple_Sensor	11/11/2021 2:19 PM	PY File	5 KB
lir_input_file	11/11/2021 1:40 PM	TXT File	1 KB



Refer to [BRTSYS\\_AN\\_002\\_LDSU IR Blaster Application](#) for more details on “LDSBus\_IR\_Blaster.py” example.

## Contact Information

Refer to <https://brtsys.com/contact-us/> for contact information.

System and equipment manufacturers and designers are responsible to ensure that their systems, and any BRT Systems Pate Ltd (BRTSys) devices incorporated in their systems, meet all applicable safety, regulatory and system-level performance requirements. All application-related information in this document (including application descriptions, suggested BRTSys devices and other materials) is provided for reference only. While BRTSys has taken care to assure it is accurate, this information is subject to customer confirmation, and BRTSys disclaims all liability for system designs and for any applications assistance provided by BRTSys. Use of BRTSys devices in life support and/or safety applications is entirely at the user's risk, and the user agrees to defend, indemnify, and hold harmless BRTSys from any and all damages, claims, suits, or expense resulting from such use. This document is subject to change without notice. No freedom to use patents or other intellectual property rights is implied by the publication of this document. Neither the whole nor any part of the information contained in, or the product described in this document, may be adapted, or reproduced in any material or electronic form without the prior written consent of the copyright holder. BRT Systems Pate Ltd, 1 Tai Seng Avenue, Tower A, #03-01, Singapore 536464. Singapore Registered Company Number: 202220043R

## Appendix A – References

### Document References

[BRTSYS\\_API\\_001\\_LDSBus\\_Python\\_SDK\\_Guide](#)

[BRTSYS\\_AN\\_002\\_LDSU IR Blaster Application](#)

Acronyms and Abbreviations

Terms	Description
IDE	Integrated Development Environment
LDSBus	Long Distance Sensor Bus
USB	Universal Serial Bus

## Appendix B – List of Tables & Figures

### List of Tables

NA

### List of Figures

Figure 1 – IDM2040 Hardware Features ..... 5

## Appendix C – Revision History

Document Title: BRTSYS\_AN\_003 LDSBus Python SDK on IDM2040 User Guide

Document Reference No.: BRTSYS\_000016

Clearance No.: BRTSYS#019

Product Page: <https://brtsys.com/ldsbus>

Document Feedback: [Send Feedback](#)

Revision	Changes	Date
Version 1.0	Initial Release	29-11-2021
Version 1.1	Updated release under BRT Systems	15-09-2022
Version 1.2	Updated HVT references to Quad T-Junction; Updated Singapore Address	22-09-2023



**BRT Systems Pate Ltd (BRTSys)**

1 Tai Seng Avenue, Tower A, #03-01, Singapore 536464

Tel: +65 6547 4827

Web Site: <http://www.brtsys.com>

Copyright © BRT Systems Pate Ltd

Application Note


BRTSYS\_AN\_003 LDSBus Python SDK on IDM2040 User Guide

Version 1.2

Document Reference No.: BRTSYS\_000016

Clearance No.: BRTSYS#019

**Documents / Resources**

	<p><b><a href="#">BRT Sys AN-003 LDSBus Python SDK</a> [pdf] User Guide</b></p> <p>AN-003, AN-003 LDSBus Python SDK, LDSBus Python SDK, Python SDK, SDK</p>
--	---

**Manuals+**